Predicting Heart Disease Risk Using Machine Learning

Vladyslav Shashkov, Yann Ravot, Andronikos Loukaidis

***Abstract*—This work is focused on the task of predicting the likelihood of a person developing cardiovascular diseases based on the lifestyle and medical characteristics of the individual.**

I. Introduction

We are trying to predict the coronary heart diseases. As most of the patients are not likely to develop cardiovascular diseases, the model that outputs negative predictions with a high probability is very effective according to the general accuracy. However, we also care about the positive predictions. As such, we are also computing the F1-score as an additional (and a more important for the classification problem) metric. The best-performing model in terms of F1-score has 86% predictions of negative predictions and 14% predictions of positive predictions, reflecting the dataset where approximately 1/10 patients are likely to develop cardiovascular diseases. We perform 100 iterations. If the learning rate is taken too large in the range of model convergence, we risk to overfit, as the number of negative predictions is much greater than the number of positive predictions. If the learning rate is taken too small, then we risk to undertrain the model and as such to obtain the unreliable predictions for both positive and negative labels.

Methodology

Data Preprocessing

Data preprocessing involved handling missing values, scaling features, and removing outliers. For each feature, missing values were replaced with the mean, and feature scaling was performed to normalize the data. More concretely, we removed the first 24 columns except for three columns containing gender and adult parameters, as they describe non-medical parameters such as telephone numbers. Next, we removed columns with less than 80% of data present, which eliminated around half of the total columns. (Note that if we modify this threshold to 30%, we only add around 35 columns, though they remain unreliable.) We removed values below the 1st percentile and above the 99th percentile to handle outliers. Indeterminate values were substituted by the mean of the respective column. Finally, we scaled the data to normalize it, removing any constant columns along the way.

Models Implemented

The following machine learning models were implemented:

- **Linear Regression (GD and SGD)** - Linear models using gradient descent and stochastic gradient descent.
- **Least Squares and Ridge Regression** - Regression methods using normal equations, with ridge regression incorporating regularization.
- **Logistic Regression and Regularized Logistic Regression** - Classification models optimized via gradient descent, with regularization in the latter to prevent overfitting.

Since we aim to predict binary labels, we used regularized logistic regression for classification.

Model metrics

The following model metrics were implemented:

- **Cross-validation score** - Evaluates the probability of correct prediction on the test set of the model by splitting the dataset into 3 equal-sized shuffled rows parts, composing 2 parts into a training set and the remaining part plays the role of the test set.
- **F1 score** - Evaluates the F1 score jointly with the 3-fold cross-validation score, calculating the precision and the recall over the test set.

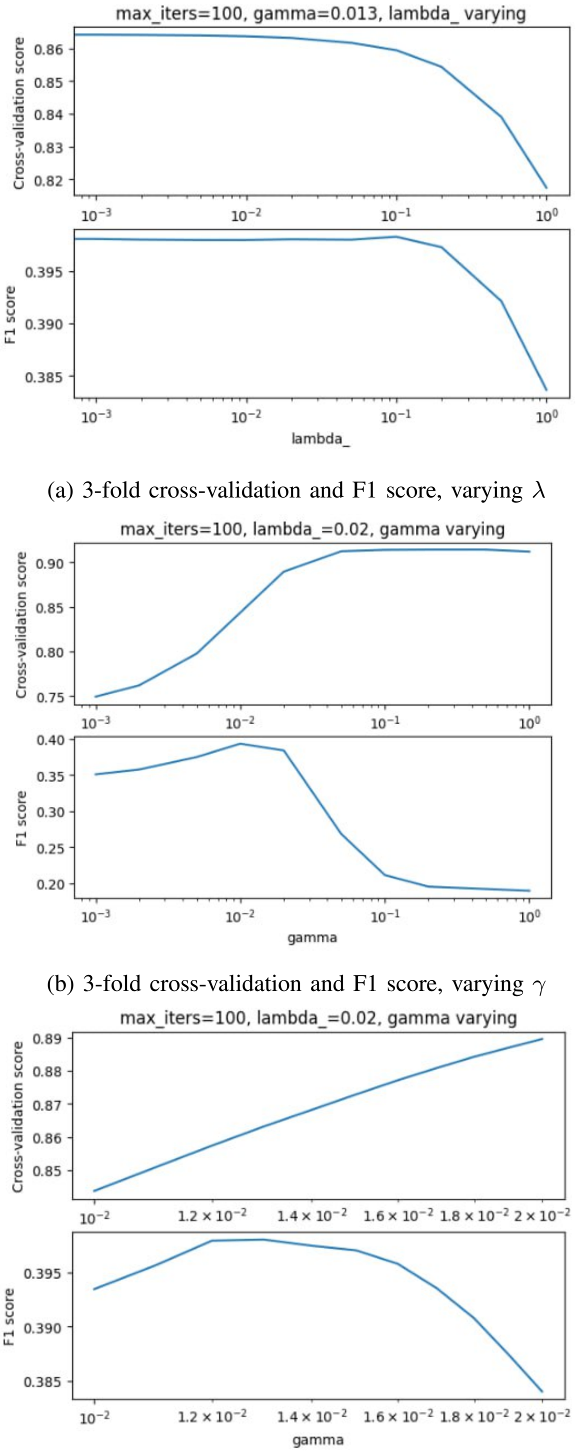
The first metric provides us with the knowledge whether the model converges or not for the given learning rate, and while it is important to keep it large for a successful model, it does not accurately describe the wellness of the model on the positive predictions.

The second metric, however, describes the performance of the model on the positive predictions, comparing true positives with false negatives and false positives, as such it is arguably a more important metric as long as the cross-validation score is reasonably high.

Hyperparameter Tuning

Hyperparameters, particularly the learning rate γ and the regularization term λ, were tuned using cross-validation. This process helped to optimize model performance while avoiding overfitting.

RESULTS



(a) 3-fold cross-validation and F1 score, varying λ

(b) 3-fold cross-validation and F1 score, varying γ

(c) 3-fold cross-validation and F1 score, fine-tuned γ range

Fig. 1: Hyperparameter Tuning Results

Figure 1a shows the effect of varying λ on cross-validation and F1 scores. As λ increases, the model's performance decreases, indicating that a low regularization is preferable.

In Figure 1b the learning rate γ is varied. The cross-validation score peaks at moderate values of γ, after which it starts to decline, likely due to the model overfitting and being more biased towards negative labels, even though the model still converges.

Figure 1c provides a closer look at γ values around the optimal range. A peak in F1 score is observed, reinforcing the choice of $\gamma = 0.013$ for best performance.

DISCUSSION

The results indicate that regularized logistic regression with a low λ and a learning rate around 0.013 provides the best performance. This balance reduces overfitting while maintaining a high prediction accuracy, attaining the F1 score of 0.403. Notably, on the aicrowd test data, it performs slightly worse than 0.015 learning rate with the same parameters, which attains F1 score of 0.405.

The data cleaning improved our model's performance quite a bit. First of all, the removal of columns with many undefined values did not worsen the F1 score of the model and improved the runtime. The removal of values above 99-th percentile and the values under 1-st percentile was also impactful, as the model's recall improved (in the case of fully converged model, the recall went from 5% to 12%, with precision of around 57% staying the same).

CONCLUSION

This project successfully demonstrates the application of machine learning methods to predict heart disease risk. Logistic regression with regularization and a tuned learning rate achieved the best performance. Removing the columns with more than 20% of information absent improved the runtime, and removing the values under 1-st percentile and above 99-th percentile got rid of outliers, improving the recall and not changing the precision. Future improvements may include using external libraries for more sophisticated models.